ELSEVIER

European Journal of Operational Research 130 (2001) 202–213

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Theory and Methodology

Implementation of a university course and examination timetabling system

M. Dimopoulou *, P. Miliotis

Athens University of Economics and Business, 76 Patission Street, Athens 104 34, Greece

Received 14 October 1998; accepted 17 January 2000

Abstract

This paper reports the design and implementation of a PC-based computer system to aid the construction of a combined university course–examination timetable. The specific difficulties to be faced are the restricted availability of classrooms and the increased flexibility of the students' choices of courses, which makes the problem very tight. The system uses an integer programming (IP) model that assigns courses to time slots and rooms. The model is coupled with flexible front-end device that generates constraints corresponding to assumptions specified by the user and report writers that facilitate the presentation of the resulting schedule. The quality of the schedule produced depends on the relative position of the courses assigned to the available time periods, a condition that the IP model attempts to satisfy by constructing groups of courses that are assigned to groups of time periods. Further, the objective function is used in a way that exploits the user's experience and knowledge of the problem. The solution of the course-timetabling problem is used to construct an initial solution to the examination timetable. A heuristic algorithm is generated to further improve it till a good feasible solution is reached. The whole system is flexible and allows the easy construction and testing of alternative schedules which are pre-conditioned according to requirements specified by the user. The Athens University of Economics and Business have used the system with success. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Timetabling; Decision support system; Mathematical programming; University

1. Introduction

The university timetabling problems deal with the scheduling of the teaching program. Two different but related problems arise in this context. One is to schedule courses and the other is to schedule examinations in the most efficient way.

Timetabling problems have attracted the continuous interest of researchers mainly because they provide the opportunity of testing combinatorial solution methods in formulations that represent difficult practical problems.

* Corresponding author. Tel.: +30-103-01-8203132; fax: +30-103-01-8226105.

E-mail address: dimop@aueb.gr (M. Dimopoulou).

PII: S 0 3 7 7 - 2 2 1 7 (0 0) 0 0 0 5 2 - 7

Graph coloring, heuristics, integer programming (IP), neural networks, tabu search, genetic algorithms, knowledge based methods, and constraint logic programming are among the approaches that have been proposed in the literature. Some of the most representative published papers include: Aubin and Ferland [4], Barham et al. [6], Johnson [27], Wright [35] and Alvarez-Valdes et al. [2], who have attempted heuristic methods. Balakrishnan [5], de Werra et al. [17] and Cangalovic et al. [8] have used graph coloring approaches. Abramson [1], Downsland [23], and Dige et al. [20] solved the timetabling problem by simulated annealing. Deris et al. [19] presented recently a case study, which applies constraint-based reasoning to university timetabling problem. Hertz [26], de Werra et al. [16] and Costa [14] have tested tabu search methods. Mulvey [32] solved the university course-scheduling problem using network flows. Mathematical programming methods have not been tried exhaustively due to the prohibitive size of the problems. Tripathy [34] and Arani et al. [3] have tried Lagrangean relaxation to solve either the course or exam-scheduling problem whereas Dimopoulou [21] and Birbas et al. [7] solved the university course and the school timetabling problem, respectively using IP. Knowledge based systems have been used to solve the university class scheduling problem [31]. More recently, decision support systems (DSSs) [28] have been developed. Optimization methods are incorporated to some of them [22,24,25]. Many of the above approaches are summarized in the review papers of Carter [9] and of Carter et al. [10,11], de Gans [18], Schmidt and Strohlein [33] and more recently of de Werra [15]. Two recent surveys by Carter and Laporte [12,13] give a complete description of timetabling methods and practices.

In most of the attempted solutions of either the course or examination problem, the objective is to find a feasible schedule. A feasible schedule is one that satisfies the teaching or examination requirements, respectively. These requirements appear usually as explicit constraints in the IP formulation while additional case specific constraints arise as a result of the particular institution's rules, administrative policies and pre-specified preferences. Constructing a feasible schedule is a difficult problem whenever there is a scarcity of classrooms and increased flexibility in the students' choices.

A more difficult problem is to produce a good feasible schedule. A good (or fair) schedule is one that has convenient relative time positions of the courses or examinations corresponding to every group of students following the same compulsory courses, that is, a compact schedule. In the present study, these requirements are faced by properly structuring the problem and by using suitable objective function coefficients in the IP formulation.

The present study describes the development of a system producing good (or fair) course and examination timetable schedules for the Athens University of Economics and business. The system incorporates both IP optimization and heuristic procedures, which are linked to flexible front-end techniques for recording the basic data of the problem, generating the constraints according to specified parameters, and to report writers presenting the 0–1 solution produced by the optimizer, in a form useful to the university management.

The presentation is organized as follows: Section 2 describes the problem addressed by the present application and the solution of course timetable by mathematical programming methods together with ways of structuring the problem in order to achieve solutions of better quality. Section 3 describes the front- and back-end devices. Section 4 reports the computational results. Finally, Section 5 presents the heuristic procedures that create the examination timetable and Section 6 states the conclusions arising from the computational experiments and the experience obtained from using the system.

2. Course timetable

2.1. Problem definition

The problem of constructing course timetables for academic institutions consists of allocating the set of courses offered by the university to time periods and classrooms in such a way that no teacher, student or room is used more than once per period and that room capacities are not exceeded. The problem addressed here deals with the assignment of each course to a number of time periods (usually hours) for a university undergraduate program with about 7000 students. The timetable covers 60 teaching periods per week. (Monday–Friday 9 a.m.–9 p.m.) Availability of large classrooms is limited and this is the main reason for extending the working hours from 9.00 a.m. to 9.00 p.m.

There are six departments in the University, each one including three or more different specializations. The academic year is divided into two independent semesters (winter and spring) each containing completely different courses. The minimum duration of undergraduate studies is eight semesters, and each semester includes some compulsory courses, and a number of optional ones depending on the specialization followed. Some of the courses are common to different departments and specializations. About 180 courses must be scheduled in each semester. According to the program of studies each course requires four hours (usually two two-hour time periods) per week. Students have complete freedom in taking compulsory and optional courses in any semester within their four-year course of studies. The large number of courses offered, together with the increased flexibility in the students' choices, result in a difficult problem with more than 10,000 variables and numerous complicating requirements, a fact that makes the feasibility of the timetable a very difficult if not an impossible task. To facilitate the construction of a fair schedule, the university administration supports the construction of a conflict free schedule only for *certain* university streams. A *university stream* is a set of compulsory and optional courses suggested by the administration to be followed by the students in each one of the eight semesters.

The major benefits of the system, described in more detail in Section 4, include: improved schedules for students, better room utilization, satisfaction of teachers' preferences, and a quick, easy and flexible way of producing and evaluating changes. All of the test problems were solved easily on a PC-type microcomputer in a Windows environment.

2.2. Problem formulation

The formulation that follows is based on the concepts of two different groupings:

- the grouping of courses, and
- the grouping of time periods.

The first defines groups of courses called *subject groups*. The courses in a subject group are followed by the same students and therefore must be scheduled at different time periods. The latter defines the *time groups*, which in the context of the specific application are four-hour groups spread into two two-hour periods. The courses assigned to a time group define a cluster, that is a group of courses scheduled on a particular time period.

The grouping operation is not new in the literature. Tripathy [34] defines student and subject groups consisting of courses offered by the university and followed by the same students; Abramson [1] defines groups of requirements consisting of a set of classes that are always scheduled together in the same period; Hertz [26] and Aubin et al. [4] define and solve the grouping subproblems, consisting of students following the same course sections; Mulvey [32] groups together classrooms with approximately equal capacities; Laporte [30] proposes a grouping of exams.

For the specific application, the formation of the subject groups was made by grouping together the compulsory courses of each university stream that are not common to other university streams. The optional courses or the courses offered to different university streams form a subject group of one course. Thus, each subject group may include from one to ten courses. A simple example showing the formation of the subject groups is the following.

Let us assume that a university stream formed by the compulsory and the optional courses offered to the students of the first semester of the department of Informatics as shown in Table 1.

Then, the subject groups formed are shown in Table 2.

Subject groups form a partition of the set of courses offered by the university. All courses in a subject group are in conflict because they are followed by the same students. The students attending a subject group may also attend the courses of

Table 1
An example of a University stream

Name	Compulsory/Optional	Common to other Departments
Calculus	Compulsory	No
Probability and Statistics	Compulsory	No
Operational Economic Analysis	Compulsory	No
Introduction to Computer Science	Compulsory	Yes
Introduction to Business Administration	Compulsory	No
Introduction to Private Law	Optional	Yes
Accounting	Optional	Yes
Introduction to Marketing	Optional	Yes

Table 2
The subject groups

Subject group name	Courses included
1	Calculus, Probability and Statistics, Operational Economic Analysis, Introduction to Business Administration
2	Introduction to Computer Science
3	Introduction to Private Law
4	Accounting
5	Introduction to Marketing

some other subject groups. Therefore, some subject groups may be in conflict. In the above example, the compulsory and the optional courses of a stream form different subject groups, which are followed by some common students. Conflicts between subject groups are represented by the *conflict matrix*. The conflict matrix is initially a square matrix including for each subject group the subject groups in conflict with it. Each row i of the conflict matrix designate the subject groups that are in conflict with subject group i by assigning the value of 1 to the corresponding column. The final form of the conflict matrix is not square since identical rows have been deleted to eliminate identical constraints. The construction of the conflict matrix helps in formulating the constraints that no student must attend more than one course at the same time.

Course conflicts for teaching staff have not been included among the constraints since approximately 50% of the academic staff teaches two different courses per semester, while the other 50% teaches only one. Conflicts may be avoided by specifying the teacher's preferences for each subject. Moreover, the specified preferences may serve to create an acceptable schedule for every teacher.

Each course, generally, needs four hours per week in the timetabling schedule. The 60 working hours of the week have been partitioned in 15 four-hour time groups, in such a way as to follow the existing timetabling practice.

The time groups formed for the specific application are shown in Table 3.

The grouping operation provides the following benefits:

- The resulting problem has fewer variables and constraints since the conflicts among the courses

Table 3
The time groups

Hours	Time groups				
	Monday	Tuesday	Wednesday	Thursday	Friday
9.00 a.m.–11.00 a.m.	1	4	13	1	4
11.00 a.m.–1.00 p.m.	2	5	14	2	5
1.00 p.m.–3.00 p.m.	3	6	15	3	6
3.00 p.m.–5.00 p.m.	7	10	7	10	13
5.00 p.m.–7.00 p.m.	8	11	8	11	14
7.00 p.m.–9.00 p.m.	9	12	9	12	15

included in a subject group are automatically satisfied. The problems resulted by applying the method to the Athens University of Economics had on average 1000 variables and 500 constraints.
- The proper formation of the time groups usually results in good solutions in that:
  - the idle periods in the students' schedule are reduced in number,
  - each course is not assigned in continuous periods, but rather, it is evenly spread over the week,
  - establish clearer morning–afternoon and evening sessions, and
  - large audience courses are assigned to periods most convenient for the students.
- The timetable produced is easily memorizable. For example, one can easily remember that the course taught every Monday 9.00–11.00 a.m. is also taught every Thursday 9.00–11.00 a.m., or that another course follows the first one and so it is taught every Monday and Thursday 11.00–1.00 a.m.

The classrooms have been partitioned into six groups of different capacity. The rooms in each group are considered to be of equivalent capacity. Each subject group can be assigned only to a group of classrooms of adequate capacity.

The IP formulation, which involves assignment of subject groups to a number of time groups, is as follows:

Notation

Sets

I	set of all subject groups
J	set of time groups
R_l	subset of subject groups that can be allocated to classroom group l
L	set of values of l
T_m	subset of subject groups in conflict; the mth row of the conflict matrix
M	set of values of m

Indices

i	index for subject groups
j	index for time groups
l	index for different classroom types
m	index for subsets of subject groups in conflict

Decision variables

x_{ij}	1 if subject group i is assigned to time group j, 0 otherwise

Parameters

a_l	number of classrooms of type l
$s(i)$	number of courses in subject group i

2.2.1. The model

Using the above Notation the objective function and the constraints take the following form:

$$\text{Max} \sum_{i,j} c_{ij} x_{ij}$$

subject to

$$\sum_{j \in J} x_{ij} = s(i) \quad \forall i \in I, \tag{P1}$$

$$\sum_{i \in R_l} x_{ij} \leqslant a_l \quad \forall j \in J, \ \forall l \in L, \tag{P2}$$

$$\sum_{i \in T_m} x_{ij} \leqslant 1 \quad \forall j \in J, \ \forall m \in M, \tag{P3}$$

x_{ij} is 0–1 $\quad \forall i \in I, \ j \in J.$

Constraints P1 assign each subject group i with $s(i)$ courses to exactly $s(i)$ time groups. This automatically takes care of the fact that the subjects in a subject group which are in conflict are assigned to different time groups and thus the number of conflict constraints is reduced. Constraints P2 take care of the classroom availability and constraints P3 ensure assignment of at most one subject group of the set of subject groups in conflict to a time group (so that for each time group at most one of the subject groups in conflict is assigned).

2.2.2. Objective function

The value of coefficient c_{ij} expresses the desirability of the assignment defined by variable x_{ij}. If all the coefficients c_{ij} are 0, then a feasible solution – if one exists – is produced. In some tests the c_{ij} coefficients were designed to induce a compact schedule by guiding, through high c_{ij} values, the assignment of compulsory and optional courses belonging to a stream to specific neighboring time groups. This to some extent offsets the inability of the IP model to take care of the desirable relative position of the subjects.

3. The complete system

The developed complete system enables the user with pop-up menus to enter, modify and display all the appropriate data. It also includes automatic computational procedures, which perform all data-processing tasks necessary to construct the subject and time groups, to create and solve the mathematical programming model and to generate the reports. Finally, the generated solution is evaluated according to criteria provided by the user.

The complete system was implemented on a PC type microcomputer (Pentium III processor) with Microsoft Access '97. The IP model was solved with both MPCODE package (DOS version) [29] and XPRESS-MP for Windows [36]. Data from the Athens University of Business and Economics were used.

The system consists of five modules as shown in Fig. 1.

The Data module includes the raw data and all the tables of the data base system such as: the courses with their associated information (name, department, semester, university stream, teacher, hours per week, room size, number of students, etc.), the time groups, the room groups, the rooms with special characteristics, the faculty members' teaching load and the specializations of each department with their different characteristics. The user can easily handle and modify, if necessary, all data involved. Appropriate parameters have been defined to distinguish the courses of the winter semester from those of the spring semester, the number and kind of time groups and the maximum number of courses in a subject group. Preassignments, specified by the user, are possible only for subject groups of one course by fixing the corresponding x_{ij} variable to one. Moreover, availabilities and preferences of each faculty member are incorporated in the data and may be easily modified. Data are displayed in tables or in forms.

The Control System module includes the user's interface with the system and the matrix generator of the IP model. The 180 courses are partitioned into approximately 70 subject groups. Only 10% of the subject groups include more than one course. Different specifications have been created for the foreign language courses and for the lab courses that accompany certain main courses.

The Optimization module includes the IP code. The resulting solution may be modified either by the user manually, or through modifications in the initial data or in the parameters. In the last case the problem is resolved with the new data.

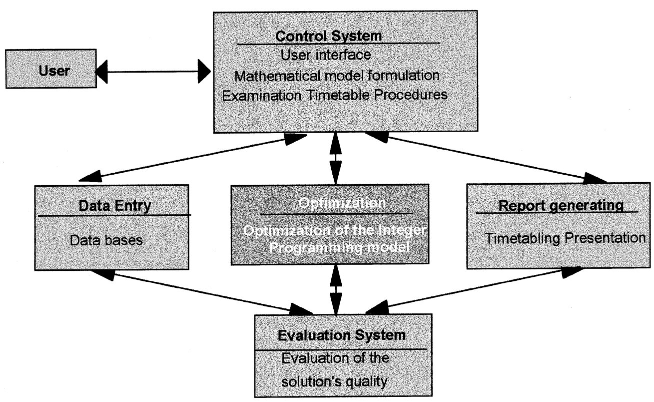

Fig. 1. The complete system.

The Report Generating module produces the following reports:
- the course timetables for each department and stream separately;
- the classroom load timetable;
- the teacher's schedule; and
- the courses scheduled for every time period of a given day for each department and stream separately.

The Evaluation Module examines the quality of the timetable produced. The evaluation of the schedule takes place in a semi-automated way. The administration unit responsible for the program collects a set of preferences that comply with some general rules for constructing the program (e.g. morning periods for compulsory subjects or a preference within the pre-specified time groups). Depending on the importance of the preference expressed the program places a high or medium value in the corresponding objective coefficient. In practice, the solution produced satisfies a high proportion of the preferences. In case that the solution produced fails to satisfy a sufficient number of preferences or in case that the solution happens to have serious drawbacks, the user resorts to the construction of alternative scenarios.

4. Computational results

The implemented system has been tested and used by the Athens University of Economics and Business since 1995. The system is also used for exploiting ways of allocating rooms to departments.

4.1. The data

The data collected correspond to two different problems one for the winter and one for the spring semester, each with approximately 1000 variables and 500 constraints.

4.2. Tests

Extensive tests were conducted on variations of the above two problems produced by changing the number and kind of time groups, the maximum number of subjects in a subject group and the objective function's coefficients.

4.3. Results

In most tests the optimal solution was found in less than one minute time. Results showed that the particular problem though NP-complete in practice was not difficult to solve. This is probably due to the fact that constraints P1 (the core of the problem) are of the transportation type with 0–1 variables and although the presence of side constraints P2 and P3 destroys the unimodular structure, fractional solutions are easily eliminated because constraints P3 act as very strong cuts. However, the larger the number of non-zero coefficients in the objective function the longer it took to reach the optimal solution.

The preferences, specified by high objective function values were almost completely satisfied in most of the tests with sparse objective functions. About 80% for the more densed ones were satisfied given that no contradiction was implied by the high objective function values used to express specific preferences.

4.4. Practical considerations

- Teaching staff's conflicts were resolved by using proper c_{ij} coefficients expressing their preferences. In the few cases in which specific conflicts insisted mainly because the IP solution did not satisfy specific preferences, appropriate changes were made manually.
- The courses that required two instead of four time periods, e.g. tutorials or seminars, have been grouped in pairs and each pair has been assigned to a single time group.
- The introduction of pre-assignments decreases the feasible region since some of the pre-assigned courses may not be in conflict.
- The formation of subject groups reduces the feasible space especially when the total number of courses in a subject group is big enough so that the total number of periods needed to be sched-

uled approximates more closely to the total number of periods available. However infeasibilities occurred rather infrequently and in these cases feasibility was restored by breaking the longer subject groups into smaller ones and reoptimizing. The benefit that the subject groups offer to the quality of the schedule is estimated to offset the reoptimization burden.

- Finally, it is worth noticing that the reduction of the problem's size and the organization of the data, that the subject and time groups offer, provide considerable help in constructing a good feasible solution while any infeasibility due to grouping is easily detected and restored in most cases. The flexible structure of the data organization, due to the front- and back-end devices, provides ways of easily exploiting alternative grouping schemes as an ultimate way of attacking infeasibilities. This was necessary only in very few cases.

5. Examination timetabling

Examinations in Greek universities take place three times a year. At the end of each semester (winter and spring) there is a normal three-week examination period where all the courses offered during the semester are examined. There is also a third double-length examination period in September including all courses of both semesters. Generally, students have complete freedom in taking the exams at any exam period (in which they are scheduled) within their course of studies.

In every examination day there are four different periods. (8.00–11.00 a.m., 11.00 a.m.–2.00 p.m., 2.00–5.00 p.m., 5.00–8.00 p.m.) Since students are allowed to have increased flexibility in selecting courses, all the courses offered by a department must be examined at different periods. Even in its simplest form this task is not easy. In each department, about 30 courses are offered and therefore at least two examinations per department must be scheduled in each examination day. Either in the winter or in the spring semester, there are four university streams in each department each one including some compulsory courses.

The basic examination timetabling problem is to assign examinations to a limited number of time units (*periods*), normally lasting up to three hours, in such a way that:

1. there are no conflicts, i.e., no student is called to take more than one examination at a time;
2. all the exams are assigned to periods;
3. the seating capacity is not exceeded in any period;
4. only one compulsory course from each department must be examined in each examination day;
5. the compulsory courses of a university stream must be evenly spread over the examination period; and
6. the exam period has the smallest possible length.

The course timetable produced as described in Section 3 can be used to generate an initial solution for the examination timetabling problem. The set of courses assigned to a time group in the course timetable are conflict free and therefore can, in principle be examined during the same examination day. For example all the courses assigned to the first time group (Monday and Thursday 9.00–11.00 a.m.) are examined the first examination day. However, since examination requires increased room capacity, to generate a feasible solution, the examinations allocated to a single day are first distributed to the different time periods of the day and if necessary additional days are used to accommodate all courses so that room capacity restrictions are satisfied.

The quality of the examination timetable depends on the satisfaction of constraint sets 4, 5, and 6.

5.1. Solution method

Given the first non-feasible solution, a process is developed that brings us closer towards the satisfaction of constraint sets 1–6 above. Two main procedures were generated: procedure EXAMS, which provides the system with an initial feasible solution and procedure UFORM that rearranges the examinations in such a way so as to relax conflicts of types 4 and 5 above.

5.1.1. Procedure EXAMS

As it was mentioned earlier, the examinations of the courses assigned to a time group are assigned to one examination day. All examinations are therefore assigned to fifteen examination days and an initial, but not yet feasible, solution is formed, which is free of conflicts, since all courses assigned to a time group are not in conflict. Therefore, constraint sets 1–2 of the previous paragraph are satisfied. Procedure EXAMS distributes the examinations of an examination day to the four different periods of the day in such a way so as to satisfy constraint sets 3–4. The excessive examinations (if there exist any) are assigned to additional examination days beyond the length of the initial three-week period. These excessive assignments are then reconsidered and, in case there exist appropriate conflict-free time periods, corresponding examinations are repositioned after scanning the examination days.

To represent the exams of compulsory courses of a university stream in each examination day, we defined appropriate variables for each department l and university stream s (COM(l, s)). These variables have been defined as 16-bit-words where each bit represents an examination day having the value of 1 if an exam on that day exists or 0 otherwise. (For example, $\text{COM}(1,1) = (1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0)$ represents the assignments of the four examinations of the compulsory courses of department *1* and university stream *1* on the 16 examination days). Similarly, variables OPT(k, l) have been defined declaring the number of optional courses of each department l assigned in examination day k. Up to two optional courses of the same department could be examined per day. If the number of examination days should be increased beyond 16, then we could simply recompile the program using the appropriate data type.

Procedure EXAMS proceeds as follows:

```
PROCEDURE EXAMS
REPEAT
  Set parameters:
    FOR i = 1 to number of examinations of
    the examination day (k);
      Set parameters:
        FOR j = 1 to number of periods (4)
          ASSIGN i to period j (if con-
          straints 3 and 4 are not violated)
          UPDATE COM(l, s), OPT(k, l)
        END {FOR}
      ASSIGN i to excessive examination day
    END {FOR}
UNTIL all examination days
REPEAT
  FOR k = 1 to number of examination
  days + excessive day;
    FOR j = 1 to number of periods (4)
      ASSIGN i to period j (if constraints 3
      and 4 are not violated)
      UPDATE COM(l, s), OPT(k, l)
    END {FOR}
  END {FOR}
UNTIL all examinations of the excessive day
have been reassigned
```

This procedure results in a feasible solution. All the examinations are assigned to periods so that constraints 1–4 are satisfied. Condition 6 above which determines the quality of timetable is also examined and satisfied to an acceptable level.

5.1.2. Procedure UFORM

This procedure works on a feasible solution given by procedure EXAMS and rearranges the exams of compulsory courses in such a way so as to spread them uniformly over the examination period and therefore satisfy constraint set 5. By performing only interdepartmental rearrangements the following basic benefits are insured:

- the seating capacity is not exceeded; and
- only one compulsory course of each department is examined in each examination day.

The rearrangement is performed by using algebraic and logical expressions in the bits of the 16-bit-word already formed by procedure EXAMS.

Namely, by adding the 16-bit-words of the university streams of a department a new 16-bit-word is defined which declares the days with examinations of the compulsory courses of the specific department and the days without examinations, respectively. For example, if

$$\mathrm{COM}(1,1) = (1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0)$$
$$\mathrm{COM}(1,2) = (0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$$
$$\mathrm{COM}(1,3) = (0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0)$$
$$\mathrm{COM}(1,4) = (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 0),$$

then their sum is $\mathrm{SUMCOM}(1) = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0)$. Zero (0) in this new word declares a day which is not available for the specific department. All the unavailable days are then excluded from the rearrangement. The algorithm finds, for each 16-bit-word (COM(*l*, *s*)), the step size by dividing 16 with the number of ones that each word includes and proceeds as follows: For each 16-bit-word the procedure finds the first position of 1. If this position's distance from the first position (day) exceeds step size, then a (1) is positioned before the first (1) and in a distance equal to step size. The algorithm takes care of the unavailable positions and does not consider them as candidate positions. The positions that the (1)s take in each 16-bit-word are then characterized as unavailable and the algorithm proceeds to the next 16-bit-word.

Procedure UFORM proceeds as follows:

```
PROCEDURE UFORM
REPEAT
  Find all available positions
    FOR j = 1 to number of University streams
      Find step size
      Rearrange the (1)s
      Skip unavailable positions
    ENDFOR
UNTIL all the departments have been rearranged
```

5.2. Results

The functions have been implemented with ACCESS BASIC and the examination subsystem is incorporated in the complete system. The report generating module includes reports presenting the examination timetable for each examination day generally and for each department separately. The exact dates for each examination period are incorporated in the system through a table, so that each year the dates can be updated.

The above algorithm provides an examination timetable suitable for the examination periods at the end of each semester (winter and spring). The examination period of September is covered by a timetable that is actually the union of the timetables of the other two periods. However, since the number of students in that period is almost half of that in the sum of the other two examination periods, some examinations may be rearranged. This is done manually.

Even though no optimization is performed in the generated examination timetable, and no evaluation of the quality is made, the solution provides an examination timetable with minimum effort, taking all the appropriate data from the course timetable and using all the facilities of the complete system.

6. Conclusions

A computer based timetabling support system has been presented. The timetabling system has all the appropriate facilities for providing valuable help to the decision maker to implement a good course and examination timetable. Interactive tools are available to allow the user to modify solutions. This decision support displays all the appropriate information that helps the user to evaluate the quality of the schedule.

A mathematical programming model based on allocating subject groups to time groups was developed which results in satisfying most of the qualitative conditions. In particular the schedule obtained is characterized by reduced number of idle periods for students, better utilization of classrooms and a clearer distinction of morning and evening sessions. Teacher preferences were satisfied to a sufficient degree by using suitable objective function coefficients. Since the qualitative conditions are sometimes in conflict, the scheduler must decide which condition prefers to satisfy by properly adjusting the objective function coefficients. The grouping operations resulted in a problem of smaller size, easier to handle, simpler to solve and in more compact timetables. In some cases, the scheduler could pre-define the assignment of a course to a time group. The real world problems tested with a size of approximately 1000 variables and 500 constraints were solved in very modest computer times in a PC-environment using a Branch and Bound based computer code.

The course-timetabling solution obtained, provided a good starting point producing efficient examination timetables with minimal effort. Thus, the system presented provides an integrated application both for the course and examination timetable.

The data organization, the presentation of the results and the ease of solution procedures provide the flexibility for easily testing alternative scenarios and produce useful displays for the efficient management of institution concerning the resources.

Acknowledgements

We thank the two anonymous referees for their very useful comments.

References

[1] D. Abramson, Constructing school timetables using simulated annealing: Sequential and parallel algorithms, Management Science 37 (1) (1991) 98–113.

[2] R. Alvarez-Valdes, G. Martin, J.M. Tamarit, Constructing good solutions for a school timetabling problem, Working paper, University of Valencia, Spain, 1994.

[3] T. Arani, V. Lofti, A three phased approach to final exam scheduling, IIE Transactions 21 (1) (1989) 86–96.

[4] J. Aubin, A. Ferland, A large scale timetabling problem, Computers and Operations Research 16 (1) (1989) 67–77.

[5] N. Balakrishnan, Examination scheduling: A computerized application, International Journal of Management Science 19 (1) (1991) 37–41.

[6] A.M. Barham, J.B. Westwood, A simple heuristic to facilitate course timetabling, Journal of the Operational Research Society 29 (11) (1978) 1055–1060.

[7] T. Birbas, S. Daskalaki, E. Housos, Timetabling for Greek high schools, Journal of the Operational Research Society 48 (1997) 1191–1200.

[8] M. Cangalovic, J.A.M. Schreuder, Exact coloring algorithms for weighted graphs applied to timetabling problems with lectures of different lengths, European Journal of Operational Research 51 (1991) 248–258.

[9] M.W. Carter, A survey of practical applications of examination timetabling algorithms, Operations Research 34 (2) (1986) 193–202.

[10] M.W. Carter, G. Laporte, J.W. Chinneck, A general examination scheduling system, Interfaces 24 (3) (1994) 109–120.

[11] M.W. Carter, G. Laporte, S.Y. Lee, Examination timetabling: algorithmic strategies and applications, Journal of the Operational Research Society 47 (1996) 373–383.

[12] M.W. Carter, G. Laporte, Recent developments in practical examination scheduling, in: E.K. Burke, P. Ross (Eds.), The Practice and Theory of Automated Timetabling, Springer, Berlin, 1996, pp. 3–21.

[13] M.W. Carter, G. Laporte, Recent developments in practical course scheduling, in: E.K. Burke, P. Ross (Eds.), The Practice and Theory of Automated Timetabling: vol. 2, Springer, Berlin, 1998, pp. 3–19.

[14] D. Costa, A tabu search algorithm for computing an operational timetable, European Journal of Operational Research 76 (1994) 98–110.

[15] D. de Werra, The combinatorics of timetabling, European Journal of Operational Research 96 (1997) 504–513.

[16] D. de Werra, A. Hertz, Tabu search techniques: A tutorial and an application to neural networks, OR Spectrum 11 (1989) 131–141.

[17] D. de Werra, N.V.R. Mahadev, U.N. Peled, Edge chromatic scheduling with simultaneous constraints, SIAM Journal on Discrete Mathematics 6 (1993) 631–641.

[18] O.B. de Gans, A computer timetabling system for secondary schools in the Netherlands, European Journal of Operational Research 7 (1981) 175–182.

[19] S.B. Deris, S. Ormatu, H. Ohta, P.A.B.D. Samat, University timetabling by constraint-based reasoning: A case study, Journal of the Operational Research Society 48 (1997) 1178–1190.

[20] P. Dige, C. Lund, H.F. Ravn, Timetabling by simulated annealing, Lecture Notes in Economics and Mathematical Systems 396 (1993) 151–174.

[21] M. Dimopoulou, Solving a large timetabling problem, in: B. Papathanasiou, K. Paparrizos (Eds.), Proceedings of the Second Balcan Conference on Operational Research, Thessaloniki, Greece, 1993.

[22] J.J. Dinkel, J. Mote, M.A. Venkataraman, An efficient decision support system for academic course scheduling, Operations Research 37 (6) (1989) 853–864.

[23] K.A. Downsland, A timetabling problem in which classes are inevitable, Journal of the Operational Research Society 41 (10) (1990) 907–918.

[24] J.A. Ferland, C. Fleurent, SAPHIR: A decision support system for course scheduling, Interfaces 24 (2) (1994) 105–115.

[25] C.R. Glassey, M. Mizrach, A decision support system for assigning classes to rooms, Interfaces 16 (1986) 92–100.

[26] A. Hertz, Tabu search for large scale timetabling problems, European Journal of Operational Research 54 (1) (1991) 39–47.

[27] D. Johnson, Timetabling university examinations, Journal of the Operational Research Society 41 (1) (1990) 39–47.
[28] S.K. Kassicieh, D.K. Burleson, R.J. Lievano, Design and implementation of a decision support system for academic scheduling, Information Management 11 (1986) 57–64.
[29] A. Land, S. Powell, Fortran codes for Mathematical Programming, Wiley, London, 1973.
[30] G. Laporte, S. Desroches, Examination timetabling by computer, Computers and Operations Research 11 (5) (1984) 351–360.
[31] M.G. Martisons, K.C. Kwan, Intelligent timetabling using a microcomputer, International Journal of Educational Management 7 (5) (1993) 9–17.
[32] J.M. Mulvey, A classroom/time assignment model, European Journal of Operational Research 9 (64) (1982) 64–70.
[33] Schmidt, T. Strohlein, Timetabling construction – an annotated bibliography, Computing Journal 23 (1980) 307–316.
[34] A. Tripathy, A Lagrangian relaxation approach to course timetabling, Journal of the Operational Research Society 41 (1) (1980) 39–47.
[35] M. Wright, School timetabling using heuristic search, Journal of the Operational Research Society 47 (1996) 347–357.
[36] XPRESS – MP for Windows, Reference Manual, 1996.